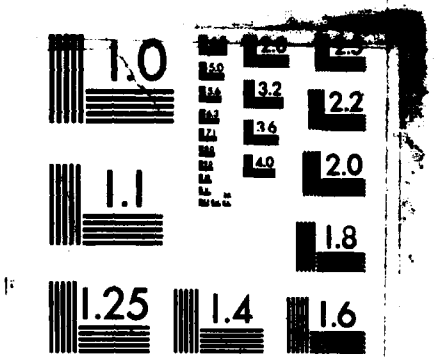
AD-A157 324 RESEARCH ON SYSTEM MAINTAINABILITY AVAILABILITY AND COST MODELS(U) POLYTECHNIC INST OF NEW YORK BROOKLYN DIV OF COMPUTER SCIENCE M L SHOOMAN MAY 84 POLY-85-010 1/1

UNCLASSIFIED N00014-75-C-0858 F/G 12/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Polytechnic Institute of New York

(2)

FINAL REPORT

RESEARCH ON SYSTEM MAINTAINABILITY, AVAILABILITY, AND COST MODELS

by

Martin L. Shooman

OFFICE OF NAVAL RESEARCH

Contract No. N00014-75-C-0858

Project No. NR 042-301

May 1982 to May 1984

Polytechnic Report No. 85-010

POLYTECHNIC INSTITUTE OF NEW YORK

DIVISION OF COMPUTER SCIENCE

BROOKLYN, NEW YORK 11201

FINAL REPORT

RESEARCH ON SYSTEM MAINTAINABILITY, AVAILABILITY, AND COST MODELS

by

Martin L. Shooman

OFFICE OF NAVAL RESEARCH

Contract No. N00014-75-C-0858

Project No. NR 042-301

May 1982 to May 1984

Polytechnic Report No. 85-010

POLYTECHNIC INSTITUTE OF NEW YORK

DIVISION OF COMPUTER SCIENCE

BROOKLYN, NEW YORK 11201

1.0 Introduction

This final report covers the work under Contract No. N00014-75-C-0858, Project No. NR 042-301, from May 1982 to May 1984. This report summarizes the major results and outlines directions for continuing research. The details of the research performed are contained in the Technical Reports and Papers listed in Section 3.0.

Most of the research reported in this section has been supported in part or in whole under this contract. However, in some cases a portion of the research was supported under other contracts and is reported here because of its relevance to the research areas of this contract.

In particular, our recent funding from NAVAIR for research on availability models of systems having built-in test and hardware and software failure modes has generated some important interactions. The basic formulation of Markov Models for system availability is being reinvestigated under this contract as described in Sec. 2.3. Also approximations and bounds for complex system availability models are being explored under NAVAIR Contract No. N00019-83-0330. (See Sec. 2.4); as are false alarm models which are discussed in Section 2.7.

The report is organized around individual sections as shown below. Section 2 contains a summary of work accomplished to date. In addition, in that section the topics for continued support are discussed. Some of these topics represent continuation of work begun under the present contract, while others represent new areas which are fruitful for investigation.

Individual Sections are:

2.0 Research Activity
2.1 Bulk Availability Analysis
2.2 Assigned Availability
2.3 Basic Availability Theory Modeling
2.4 Approximations and Bounds for Markov System Availability
2.5 Computer Aided Design for Large Markov Models
2.6 Relating the Quality Parameters of Built-In-Test-Models to Overall System Availability and Cost Effectiveness
2.7 High Level Markov Models of Built-In-Test in Systems and Devices
3.0 Publications
3.1 Published and Submitted Papers, and Reports

3.2 Talks and Seminars

4.0 Professional Activity

2.0 Research Activity

2.1 Bulk Availability Analysis

Most of the mathematical models and analyses of availability and availability (reliability) warranties have considered individual units of equipment or collections of such units in which the availability of each unit is desirable and important. One may refer to such considerations as "unit availability". The major mathematical tools for such studies are probability, statistics, and queuing theory. The advantage of probabilistic models is that a deterministic analysis of unit failure is in most cases too complex and the methodology of stochastic processes is well developed and yields useful results under many conditions. However, stochastic models have two important disadvantages: they require data for their useful implementation that are often very difficult or indeed impossible to obtain, and they are hard to interpret to people untrained in the details of probabilistic reasoning, for example as part of a contract negotiation. Thus it would be desirable to have alternative methodologies for availability models if they could be developed so as to give meaningful and useful analysis. This does not seem possible in most cases. However, for the concept of bulk availability, such an alternative methodology does seem possible. The primary goal of the work reported here was to define a non-probabilistic model for bulk availability. Such a model shows the feasibility of an alternative methodology for at least some types of availability analyses. It provides a model that is useful directly for the study of bulk availability. In particular, it gives a basis for the formulation of availability warranties in contracts that deal with bulk availability.

Bulk availability differs from unit availability in that the individual units are not the primary consideration but rather the number of units that are available at a given time. A bulk availability model applies to situations in which there are a number of similar units making up the total system which is defined as the collection of units. Thus the system itself is unstructured in the model formulation. Availability of such a bulk system is defined in terms of the units that are available. In bulk availability the individual units are not important but only the number that are in a particular state of operation at a particular time. The deterministic

model employs transfer rates between states such that the actual number of units changing state depend on these rates and the number of units in a state.

The model considers a large number of similar units that fail at a constant rate λ. Upon failure, a unit enters a waiting line and from there enters repair service. The service activity can accommodate up to r units and completes service at a constant rate μ. It is assumed that the number of units departing from a state is equal to the appropriate rate parameter times the number of units required or available depending on which state is under consideration. The model is constructed as a system of ordinary differential equations for the three quantities:

$m_a(t)$ = the number of units that are active (available),

$m_w(t)$ = the number of units that are waiting for service, and

$m_s(t)$ = the number of units in service.

All three quantities are functions of time as the independent variable.

In this model system, availability can be defined as a lower bound on $m_a(t)$.

It is assumed that the system is conservative in that $m_a(t) + M_w(t) + m_s(t)$ is a constant.

Thus the model is formulated in two regions of the (m_s, m_w) plane. Figure 1 shows the relations between the three states of the system: available, waiting, and in service.

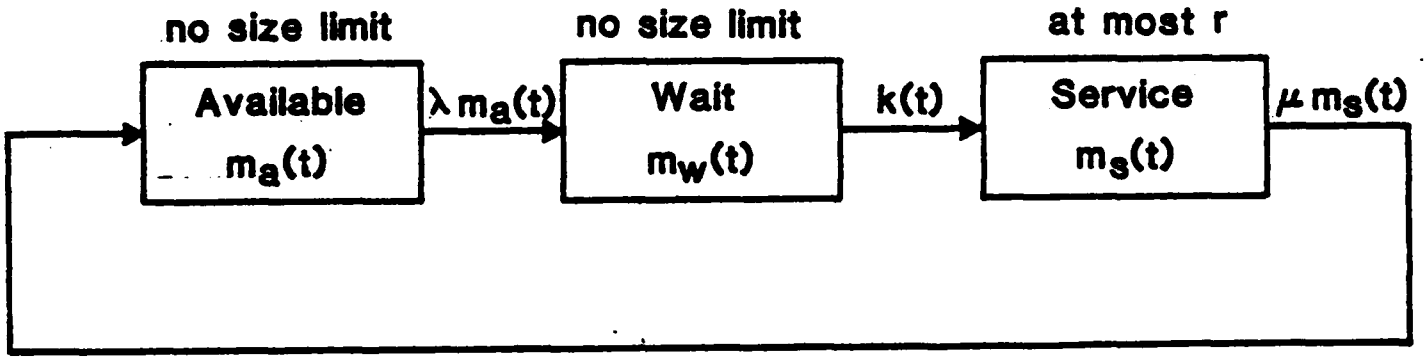


Figure 1
States of the Bulk Availability Model

The system of differential equations governing the flow of units through the states of the bulk availability model follows directly from the conservation and flow rate assumptions stated above. This system has the form:

$$\frac{dm_a(t)}{dt} = \mu m_s(t) - \lambda m_a(t)$$

$$\frac{dm_w(t)}{dt} = \lambda m_a(t) - K(t)$$

$$\frac{dm_s(t)}{dt} = K(t) - \mu m_s(t)$$

In this system, the quantity K(t) depends on the relative values of m_s and m_w leading to two specific forms for K(t) as given below. Because of the conservation assumption, the initial values $s_1 = m_s(t_o)$, $w_1 = m_w(t_o)$, and $a_1 = m_a(t_o)$ satisfy the condition $s_1 + w_1 + a_1 = m_s(t) + m_w(t) + m_a(t)$ for all values of t. Since the differential equation system must be followed across different forms the "initial" values a_1, w_1, and s_1 are incorporated as parameters into the solutions.

The value of K(t) depends on what is required by the service state and what is available from the waiting state. This leads to the following division of an (m_s, m_w) plane into two major regions.

Region 1 $K = m_w(t)$, defined by $m_s \leq r$ and $m_w < r - m_s + \mu m_s$.

Region 2 $K = r - m_s(t) + \mu m_s(t)$, defined by $m_s \leq r$ and $m_w \geq r - m_s + \mu m_s$.

These regions are shown in Figure 2.

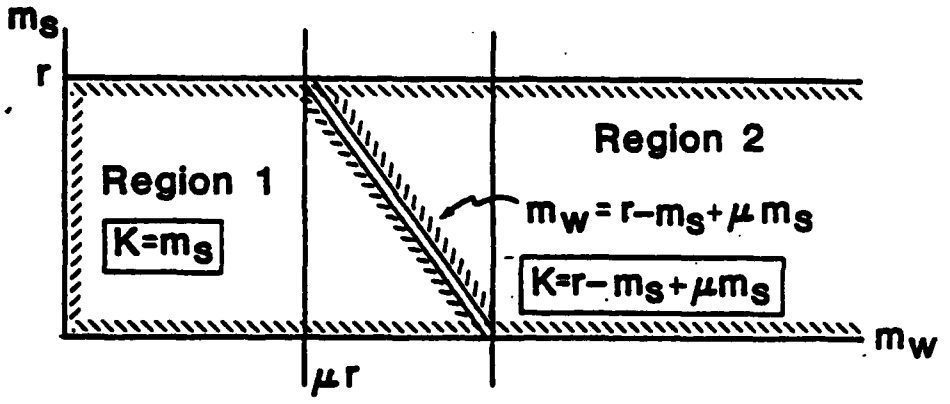


Figure 2
Regions Specifying Model Form

The system of differential equations can be solved directly in terms of explicit functions of time. The resulting functions of time are complicated and require computer evaluation.

The deterministic mathematical model for bulk availability relates the availability of a system to various system parameters. Unlike the usual stochastic model it does not depend upon an underlying probability framework and its results are specific quantities expressed as functions of time rather than expected values or other partial descriptions of random events. In this sense, the present bulk availability model is related to stochastic models in the way thermodynamics is related to statistical mechanics as a description of certain physical processes.

The value of the model is in its direct relation between a few system parameters and clearly identified descriptors of the system, particularly $m_a(t)$ which measures system availability. This direct relation is useful in contract negotiations, system design, logistic support design and implementation, and the conducting of acceptance test procedures. For the model to be meaningful, the assumptions required for the model must hold. This restricts model applicability to the bulk case in which the system description can indeed be given by a division of units into the three states of the model. Thus the model takes no account of any interaction between units or differences in utility between units. The conservation assumption also limits the model. There are only the three specified states so that

any case in which a unit fails and cannot be repaired or replaced is not included in the present form of the model. Of course such additional states can be included in the model. The desirability of doing so to extend model validity must be balanced against the desire for relatively simple system descriptors which motivated this kind of model from the beginning.

Details of the solutions to the differential equation systems for the various cases are given in: Bulk Availability Report POLY-EE/CS 83-001, February 1983 Polytechnic Institute of New York. Application of the model methodology to availability warranties is discussed in: "Warranty Definition for Bulk Availability" Proc. Annual Reliability and Maintainability Symposium (Orlando) Jan. 1983.

The complexity of model solutions required computerized procedures to be employed.

Figure 3 shows a flow chart of how the computerized model analysis is carried out. To keep track of time, three different "time like" variables are used. The variable T measures time from the start of one Region form. It changes in increments of specified amount DT.

The value of DT must be selected rather carefully because it governs the time values at which the solution forms are evaluated. If the solution remains in a single Region, DT can be relatively large, its value subject only to the detail of solution form required in a particular analysis. However, when the solution changes Regions, it should do so as near to the critical values of the solution as possible. If DT is too large, the solution from one Region will continue into the other Region on transition and the "initial values" will be incorrect being based on the previous solution form which has ceased to be appropriate at some previous time. This effect can introduce significant errors in a model analysis. A computer run of the model does not require extensive time or storage so it is feasible to try several values of DT to insure a realistic set of results. The variable TOTT records each of the times spent by the system in each solution Region. Actual time is given by the variable TLOC which is the sum of the present value of TOTT and the value of T obtained so far in the present

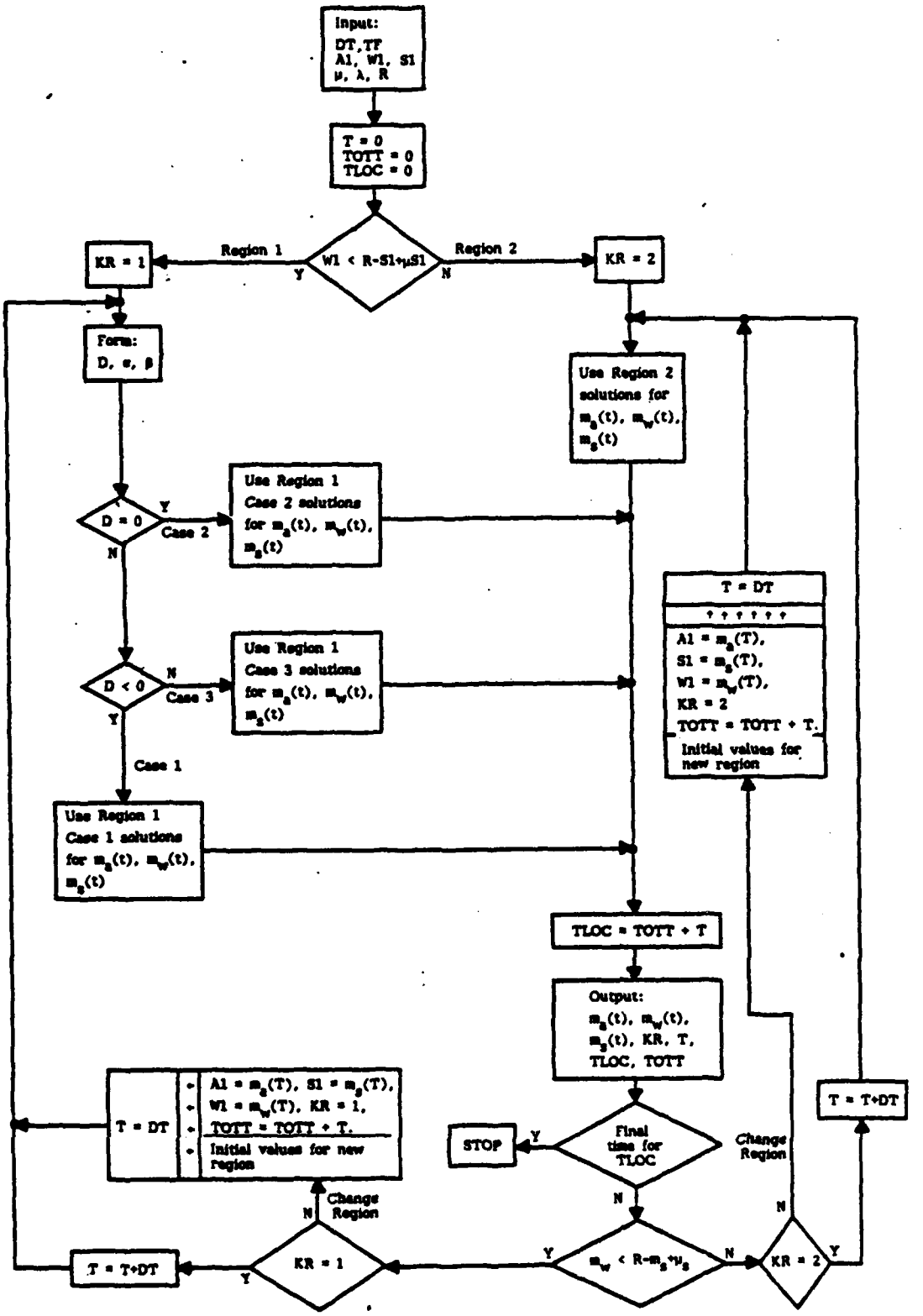


Figure 3

Flow Diagram of Computational Program

Region solution at the termination of the analysis. An analysis is terminated when TLOC reaches a value TF specified by the program. The quantities DT and TF are input values that govern the detailed calculation of the solutions, they are programming parameters rather than system parameters.

Several examples of the model are given in the Report (POLY-EE/CS 83-001). One is a simple case that does not require computational procedures, the others employ the computer program diagramed in Figure 3. One application of the model methodology in to Bulk availability warranties.

Much of the study of availability warranties has considered the availability of a single item or system which is subject to failure and repair/replace service over the duration of the warranty period. This represents a classical viewpoint in the study of availability. The present research deals with another kind of availability consideration which considers a large number of units rather than a single item.

Bulk availability warranties are defined as costs incurred by the contractor when the number of available units falls below contractually specified levels. The cost trade-offs available to a contractor are illustrated by the examples. Conclusions about the utilization and applicability of the bulk availability warranty concept have been considered as part of this research. These warranty studies are described in detail in the Report (POLY-EE/CS 83-001) and the RAMS Proceedings previously referred to.

Findings of this study to date include the following:

The bulk availability model can be employed as useful tool in trade-off analyses between manufacturing, logistic, and warranty costs. By using several versions of a proposed model, with different parameter values, a set of possible availability functions can be generated.

The deterministic model is simpler to formulate, interpret, and explain than corresponding stochastic models. This is particularly true in the present case where the appropriate stochastic model seems to be of the finite dam type in which the mathematics is more involved than for queue type models. Ease in application and in intuitive relevance between

parameter values and results is useful in contract negotiations where one party may be required to explain its analyses as part of its justification for cost trade-off decisions.

The closed form of direct solution for this kind of model allows the consideration of numerical results over any desirable time period, subject only to considerations of transition between solution regions.

The bulk availability model can be given broader interpretations. For example, if it is to be used as a model for manpower availability the quantity λ does not represent manufacturing effort, it relates to the profile of people recruited into the manpower pool.

The bulk availability model developed to date as described above can be made more widely applicable if it is extended to allow time variation in some or all of the model parameters λ, μ, and r. In the present model, the most usual form of solution is to move to Region 2 and remain there tending toward, and usually reaching rather soon, a steady state condition with the service facility full. If the parameters change with time a more dynamic model would result which would allow transitions back to Region 1. Such a dynamic model is felt to be a reasonable one in modeling situations in which a contractor can change parameter values or operating conditions cause them to change during the warranty period.

The present direction for development of the bulk availability model is to introduce two kinds of time dependent variation into the parameter λ and μ. In one extension the parameter remains constant over specified time intervals but change value for two or more intervals. Initial work will consider only two levels, defined on two or more intervals. The other variation with time will be linear over restricted time intervals.

A research topic of considerable theoretical interest is the relation of the deterministic model to stochastic models (e.g. as a limit of some kind of queuing model). Though the model seems to suggest rather direct connections to stochastic model (as observed by several people at the presentation of the model at the RAMS in Orlando, as well as by C. Marshall during model formulation) development of the connection has

MARTIN SHOOMAN (continued):

INVITED LECTURES

- o "Software Reliability and Complexity", Technion, Haifa, Israel, July 13, 1982.
- o "Computing Perspectives at the Polytechnic Institute of New York", IBM Seminar on Computing in the University Environment, La Hulpe, Belgium, Dec. 7, 1982.
- o "Managing Software Testing Using Reliability Estimates", NSIA/DOD Conference on Software Test and Evaluation, Wash. D.C., Feb. 1983.
- o "Software Reliability Estimation", ONR Research Review, Arlington, Va., April 22, 1983.
- o "Software Measurement", Bell Labs, Holmdel, N.J., Nov. 7, 1983.

SEMINARS

- o "Software Engineering", University of Tel Aviv, Israel, July 11, 12, 1982.
- o "Software Engineering", Portland, Oregon, July 27-29, 1983, Sponsored by the Society of Reliability Engineers.
- o "Introduction to Software Reliability", with Myron Lipow (TRW). Tutorial at Annual Reliability and Maintainability Symposium, January 1980 - 1984.
- o "Systems and Components Reliability", IBM, Poughkeepsie, N.Y., Dec. 9, 1983.
- o "Systems and Components Reliability", IBM, Kingston, N.Y., Dec. 16, 1983.

3.3 Research Reports

Bulk Availability, Polytechnic Institute of New York, Research Report: POLY EE/CS 83-001, Feb. 1983.

G. Carbone, T. Lovelock, C. Marshall, Tactical Aircraft Penetration Model - Final Technical Report, Fairchild Republic Company, 4/25/83, for Fighter Division ACS/Studies and Analyses, H.Q. USAF/SAGF.

Jack Kang Chan, "Stochastic Availability of a Repairable System with an Age and Maintenance-Dependent Failure Rate," Ph.D. Thesis, June 1982, and Report No. POLY EE/CS 82-004.

Behnam Ebrahimian and Leonard Shaw, "Scheduling Maintenance Operations Which Cause Age-Dependent Failure Rate Changes," Ph.D. Thesis June 1983, and Report No. POLY EE/CS 83-002.

Leonard Shaw and Samo Bozic, "Analysis Techniques to Assess Availability Improvement of Built-In-Tests (BIT)," Report No. POLY EE/CS 83-009, November 1983.

- Notes and Problems for New Course, "Selected Topics in Fault-Tolerant Computing", Fall 1982.

3.2 Talks and Seminars

CLIFFORD MARSHALL:

- "Warranty Definition for Bulk Availability", Annual Reliability and Maintainability Symposium, Orlando, Fl., Jan. 1983.
- "Graph Theory", Suffolk County Math Teachers Annual Workshop, Kings Park, N.Y., March 1984.

LEONARD SHAW:

- Short course lectures on System Reliability, U.S. Army ARRADCOM, Dover, N.J., Jan. 1983.

SAMO BOZIC

- "Analysis Techniques to Assess Availability Improvement of Built-in-Tests. ORSA/TIMS Meeting, Chicago, April 1983.
- "Computation of Transients in Large Markov Chains", submitted to ORSA/TIMS Meeting, Dallas, October 1984.
- "Is BIT a Blessing, a Toy, or an Annoyance", submitted to 1985 Reliability and Maintainability Symposium.

MARTIN SHOOMAN:

TALKS

- "Overview of Software Reliability", ITT Symposium, NYC, Oct. 15, 1982.
- "Reliability of Shuttle Mission Control Center Software", (with G. Richeson, NASA), Annual Reliability and Maintainability Symposium, Washington, D.C., Jan. 1983. (Prize Paper)
- "Teaching of Software Engineering", ACM SIGSE Technical Symposium on Computer Science Education, Orlando, Fla., Feb. 17, 1983.
- "Software Engineering Education and Research", NSIA Meeting, Crystal City, VA, April 27, 1983.
- "Applications of Stochastic Models and Graph Theory in Software Engineering", NSF/Stevens Conference on Networks, Aug. 26, 1983.
- "Overview of Software Engineering", LIFT Symposium, Melville, N.Y., May 24, 1984.

3.0 Publications

This section lists publications, talks, and seminars generated by Polytechnic faculty with support from the present contract. The material is organized by type into two subsections and by name of investigator within subsections.

3.1 Published and Submitted Papers, and Reports

CLIFFORD MARSHALL:

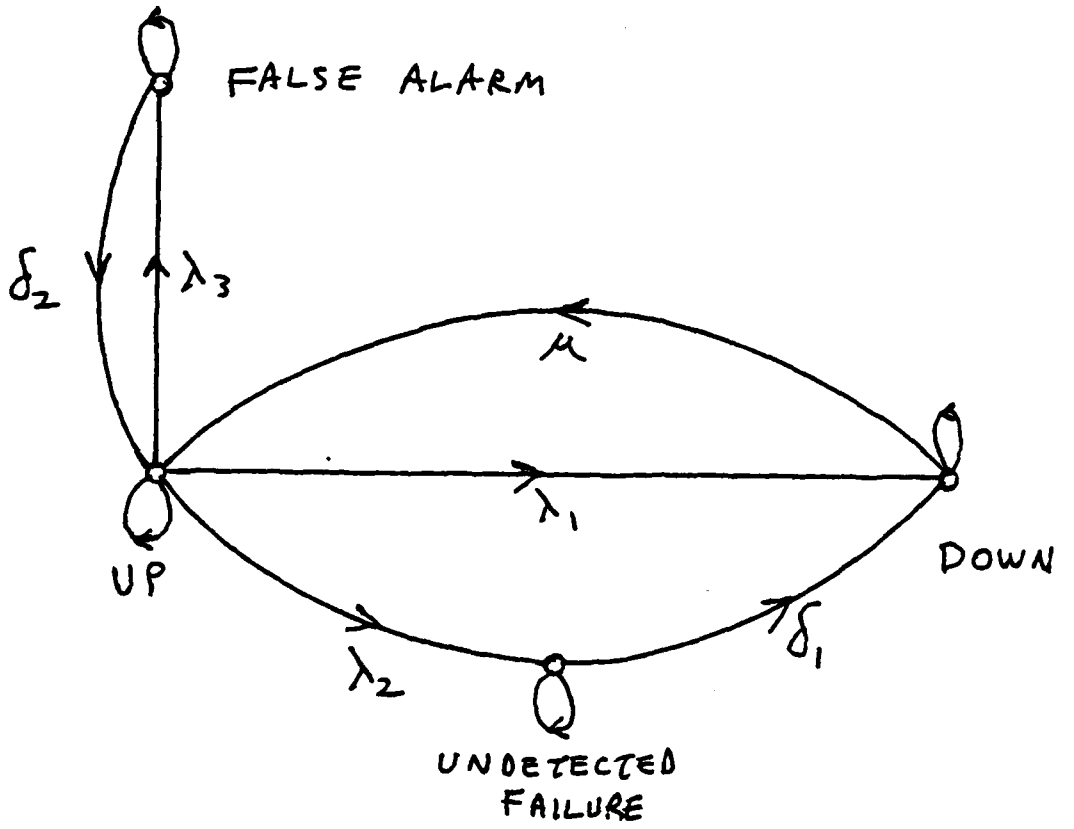
- "Warranty Definition for Bulk Availability", Proc. 1983 Annual Reliability and Maintainability Symposium.
- Book Review "The Single Server Queue, Revised Edition by J. W. Cohen", Technometrics, Vol. 25, Feb. 1983.

LEONARD SHAW:

- "On Receding Horizon Feedback Control", with C. C. Chen, AUTOMATICA, Vol. 18, No. 3, May 1983, pp. 439-352.
- "Scheduling Maintenance Operations Which Cause Age Dependent Failure Rate Changes", with B. Ebrahimian and J. K. Chan, Proceedings of 10th IFIP Conference on System Modeling and Optimization, Springer-Verlag, 1982, pp. 834-840.
- "Stable Speed Control of a Current-Fed Inverter-Driven Induction Motor", with X. T. Wang and E. Levi, Proceedings of June 1984 American Control Conference, IEEE, Piscataway, N.J.

MARTIN SHOOMAN:

- "Reliability of Shuttle Mission Control Center Software", with G. Richeson (NASA Johnson Space Flight Center), Proceedings Annual Reliability and Maintainability Symposium, January 1983. (Prize Paper).
- "Teaching of Software Engineering", ACM SIGSE Bulletin, Feb. 1983.
- "System Availability Bounds and Approximations", Submitted May 1, 1984, to 1985 Annual Reliability and Maintainability Symposium.
- "Software Reliability - Historical Perspective", Special Centennial Issue of the IEEE Reliability Transactions, June 1, 1984. (Invited Paper)
- Software Engineering: Design, Reliability, Management, McGraw-Hill, 1983.
- Instructor's Manua l for Software Engineering, McGraw-Hill, in preparation.
- Probabilistic Reliability: An Engineering Approach, Updated Edition, in preparation.



where:

λ_1 = ordinary failure rate

λ_2 = undetected failure rate

λ_3 = false alarm rate

μ = ordinary repair rate

δ_1 = detection rate for undetected failures

δ_2 = diagnosis rate for false alarms

Fig. 1 A Markov Model for Distributed BIT

2.7 High Level Markov Models of Built-In-Test in Systems and Devices

The emergence of relatively cheap and powerful computers has led to many revolutions in electronic circuitry and systems. In addition to providing immense functional computing power, there is ample additional capacity for self diagnosis of failures (Built-in Test, BIT). If computation including BIT is concentrated in a single computer complex of minicomputers or superminicomputers we refer to the system as having centralized computing. If each (or most) major element or subsystem has a microprocessor associated with it for functional computing and BIT, then we refer to the system as having distributed computing. We will discuss modeling of a system with distributed computing and BIT in this section.

The Markov model given in Fig. 1 is a high level Markov model representing the major states of a system element with BIT.

There are separate states in the model for ordinary UP and DOWN states as well as FALSE ALARM and UNDETECTED FAILURE states.

We are presently solving this model and investigating whether a set of bounds similar to those of Sec. 2.4 can be derived for a system composed of modules modeled as shown in Fig. 1 as well as ordinary modules (same as Fig. 1 with $\lambda_2 = \lambda_3 = 0$).

A similar approach to that described in the preceding paragraph will be used to model dependent hardware-software failure modes. A basic Markov module with additional coupled states will be defined and we will then investigate how to embed this module in a more complex system.

We have already begun a study of a number of systems so that we can compare the exact system availabilities with the values given by the bounds and approximations discussed above. The initial candidates for such a study are two series elements, two parallel elements, one and two elements which include built-in test, false alarm, and undetected failure states (See Sec. 2.7), and introductory hardware/software reliability models. Exact expressions are being derived for these cases using paper and pencil analysis, and these will be checked by solving with the ARIES availability modeling program. We will then compare the analytical solutions with the bounds and approximations described above.

References

Shooman, Martin L., Independence Bounds and Approximations for System Availability Computations, Research Memo No. R1, Division of Computer Science, Polytechnic Institute of New York, April 6, 1984.

Siewiorek, Daniel P. and Robert S. Swarz, The Theory and Practice of Reliable System Design, Digital Press, Bedford, Ma., 1982.

2.5 Computer Aided Design of Large Markov Models

Models having large numbers of states usually result from interconnections of several smaller submodels. We are investigating classes of interconnection patterns, and associated structural properties with a view toward ease of data entry into computer analysis programs. The structural properties are also being used to develop efficient recursive algorithms for computation of transient behavior of state probabilities. Arguments for neglecting the influence of improbable states arise naturally. This project is also investigating the automatic fitting techniques to march Markov models to empirical lifetime distributions.

2.6 Relating the Quality Parameters of Built-In-Test Modules to Overall System Availability and Cost-Effectiveness.

This project is developing interactive computer programs and nomograms to aid the designer who must evaluate the possible benefits and costs of adding built-in-test to a complex system. Reports of undesirable, and possibly unexpected, poor performance in systems with built-in-tests have motivated this study in which we concentrate on the effects of false alarms and reliability of the BIT itself on the ultimate availability of the overall system. Useful results are being obtained with minimal assumptions about the relative reliability of the BIT vs. the functional circuit, and relative repair times with and without the BIT.

$$P(Y_i') = \frac{\lambda}{\lambda + \mu} - \frac{\lambda}{\lambda + \mu} e^{-(\lambda+\mu)} \quad (3)$$

$$P(W_i) = e^{-\lambda t} \quad (4)$$

The details of the computations for this problem appear in Shooman 1984. Substitution of Eqs. 3 and 4 into the system availability function, Eq.2, yields an upper bound on the availability

$$A_{sy} <= 1 - 2P(Y_i')^2 \times P(W_i)^2 - 4P(Y_i')^3 \times P(W_i) - P(Y_i')^4 \quad (5)$$

We are presently investigating under which conditions this method yields an approximation versus a bound.

Lower Approximations to System Availability

One can also formulate a lower approximation to the system availability if one considers Eq. 1. We replace each of the UP probabilities by the reliability for the individual component, and each of the DOWN probabilities by the unavailability for individual components. Thus, since each individual term is replaced by its lower bound, the new expression represents an approximation which is sometimes a lower bound on system availability.

We will refer to the upper approximation developed above as the independence upper approximation, A_{IUA}, and the lower approximation stated in this section as the independence lower approximation, A_{ILA}.

We are presently investigating under which conditions this method yields an approximation versus a bound.

Continuation of This Research

Since we have approximated the function from above and below, we can approximate the system availability by taking the average (or a weighted average) of the upper and lower approximations.

$$A_{sy} = 1 - \sum_{i=j+1}^{i=k} P(\text{all the DOWN states}) \qquad (2)$$

Our notation is that Y_1 denotes the event, element one is up at time t, and W_1 denotes the event, no failures of element one in interval o to t. The complementary events are denoted by primes.

The Independence Upper Approximation on System Availability

To formulate the upper approximation to the system availability, we assume that all the Y_i and Y_i' events are independent and replace each event with a lower bound. Since all these terms are substracted in Eq. 2, this represents an approximation which is sometimes an upper bound on A_{sy}.

We illustrate the application of the upper bound technique to the system of Fig. 1.

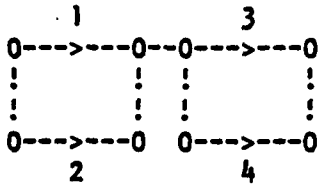


Fig. 1 Example of a Four Element System

There are seven down states for the element shown above: S_{34}, S_{12}, S_1, S_2, S_3, S_4, S_0. The notation used for the state subscripts denotes the elements which are UP, and by implication those elements omitted are DOWN. Thus, S_{34} represents the state where elements 3 and 4 are UP and 2 and 3 are DOWN; S_0 represents the state where zero elements are UP, i.e. all are DOWN.

For simplicity in this example, we will assume that all the components have the same failure rate, λ, and the same repair rate, μ.

Thus, the individual unavailabilities and reliabilities become:

The research described in Secs. 2.4, 2.5, 2.6, and 2.7 provides some new approaches to these problems.

References

Bavuso, Salvatore J., A Users View of CARE III, Proceedings, Annual Reliability and Maintainability Symposium, IEEE, N.Y., 1984, pp. 382-389.

Doyon, Leonard R., and Martha W. Berssenbrugge, Solving Complex Reliability Models by Computer, Proceedings, Annual Reliability and Maintainability Symposium, IEEE N.Y., 1968, pp. 431-448.

Flemming, Randall E., and Linda Jo Dolny, Fault-Tolerant-Design-to-Specs with GRAMP & GRAMS, Proceedings, Annual Reliability and Maintainability Symposium, IEEE N.Y., 1984, pp. 403-408.

Shooman, Martin L., Probabilistic Reliability: An Engineering Approach, McGraw-Hill, N.Y., 1968.

2.4 Approximations and Bounds for Markov System Availability Models

The work described in Research Memo R1, Shooman 1984, establishes an upper and lower bound on the system availability based on the well-known and relatively simple expressions for the availability and reliability functions of a simple component. Thus, we no longer have to solve a set of differential equations, but have reduced the computation to a combinatorial problem.

Our purpose is to develop an upper and lower bound on the system availability time function based upon a set of probabilities associated with each of the system elements. These element probabilities will be relatively simple to calculate (compared with complete solution of the system availability model), and combining the upper and lower bounds yields a useful approximation to the system availability.

Formulation of the System Availability

One can formulate the availability function of the system by focusing on either the probability that the system is up or the probability that the system is down. If the availability model has j UP states and k DOWN states, (j + k = m), then we can write the system availability function, A_{sy} as a sum of state probabilities, since all the states are disjoint

$$A_{sy} = \sum_{i=1}^{i=j} P(\text{all the UP states}) \qquad (1)$$

or

make looking at a variety of arrangements and comparing their properties feasible. The research on assigned availability will do this while continuing to study general properties of double correspondence graphs.

2.3 Basic Availability Theory Modeling

The standard technique for studying the availability of a repairable system is to begin by formulating a Markov probability model for the system (Shooman 1968, Sec. 2.8, 5.8.2, 6.10.2). The solution of such a Markov model for a system of n elements (components, subsystems, replaceable units, modules, etc.) involves m system states, where $m <= 2^n$. The various state probabilities are computed by writing a system of coupled first order differential equations and solving them for the m system states. The system availability, A, is then computed by merely summing the probabilities, (all Markov model states are mutually exclusive), for the subset of system states where the system is UP. Similarly, the unavailability, U = 1 - A, can be obtained by summing the probabilities for the system DOWN states.

Modern systems are extremely complex with central computers, digital sensors and distributed microprocessor controllers, and built in test BIT equipment. The high levels of availability required in modern computers have given rise to sophisticated redundancy and recovery techniques characterized as Fault-Tolerant computing. The inclusion of such factors introduces additional complexity in the models. Furthermore, most modern digital electronic equipment incorporates BIT, which requires that we model additional system states which represent false alarms and undetected failures. Along with the array of computers in modern day systems comes a pleothera of software to perform the various computational and control tasks, since the software can also fail (due to the excitation of undetected latent errors), it also must be modeled. Lastly, if we are to realistically characterize the logistics aspects of repair we must include the number of available spares, the waiting time for shipping from a depot, etc. All the above factors greatly increase the number of states in the system model. For any sizeable problem, n = 10 or 20, m = about 1,000 or 1,000,000, and even modern large computers require very long solution times. Furthermore, due to the complexity of the problem, there is little design insight, without extensive computer runs to establish sensitivities, as to how the various parameters effect the solution.

by expressing a cycle as a base ten number so that each cycle required only one location rather than M. However, H is still large for most graphs of interest.

Another limitation of the present program is that it was developed for non-multiple edge graphs. However, some interesting arrangements employ multiple edges (e.g. the lower bounds on full service probability obtained by Marshall employed a multiple edge graph).

Continuation of this work will extend the analysis to include multiple edges, investigate other schemes (hash coding) for storing the list of cycles, and generate NC(k) from the cycles. An increase in storage availability (in the near future) will allow consideration of more graphs so that comparison studies can be made.

The multiple correspondence concept seems applicable to other areas such as some aspects of fault tolerant computer design. In particular the architecture of multiprocessor systems in which units of cache memory can be assigned (dedicated) to different processors can be directly modeled as a multiple correspondence.

Analysis of such models, however, continue to be difficult. One additional direction presently under study emerges from the consideration of the components of a graph after the restrictive event (removal or inoperation of edge "units") occurs. Any component that is a tree will not have full service. Any component that has at least one cycle will have full service. This consideration can be studied by means of a computer program, presently under development, that identifies the components and determines if they have cycles. The edges in a component that form more than one cycle are redundant in the context of assigned availability and as such are an extra cost.

The ideal of course is to have an arrangement that maximizes the number of components that have an extra edge (to form a cycle) and minimizes the number of redundant edges. This could be in the expected value sense, bringing in a requirement to look at the variance values involved; or it could be expressed as some type of probability requirement. The difficulty is in getting methods of analysis of models like these. The availability of computer power and programs under development make

in that they contribute to the formation of a larger number of cycles. To study this question consideration was given to the generation of a vector N C (k) whose components give the number of cycles upon which an edge k is to be found. Such a vector may be used as a tool in searching for particularly significant cycle bases, or in testing a specific cycle basis for effectiveness in cycle generation.

Initial work on developing an algorithm to obtain NC(k) from the graph incidence matrix was found to be inefficient and some new directions were taken. In the present state of research NC(k) can be obtained as part of the general study of cycles.

Because cycles seem to play an important part in the study of assigned availability and are difficult to deal with unless they are under some kind of control, a computer program was developed to study the cycles in a graph. The development was in two major parts:

(a) A cycle basis is formed for the graph based on the (unique) depth first spanning subtree of the graph. If the graph is not converted, the spanning sub forest is used.

(b) The cycle basis is used to generate the set of all cycles in the graph that use an edge at most once. Some of the cycles are simple, using each vertex only once and these are not simple. The simple and non-simple cycles are identified and enumerated.

In this work the graph is represented by a pointer array that gives the adjacency structure of the graph. Edges are numbered by the program which also makes available the vertex pair associated with each edge.

As cycles are generated by linear combinations of basic cycles it is possible to generate the same cycle in different ways. Therefore, all distinct cycles must be retained as tests for new cycles. Since the number of cycles becomes large for even small size graphs this storage is a problem. In particular the research to date employed an IBM PC with limited storage. Initially the cycles were stored as M dimensional arrays where M is the number of edges. Thus if there were H cycles the cycle list required HM storage locations. An improvement was obtained

is the expected number of tasks able to receive assigned units after the random removal of units. Another measure is the probability that all tasks can be assigned units, this has been called full service probability by Marshall.

In the case k = 2 a multiple correspondence can be represented as a linear graph called a double correspondence graph (Marshall). In this representation the vertices stand for tasks and the edges are units. The incidence of vertices and edges specifies the association between tasks and units. A unit can be assigned to either of the tasks (vertices) with which it is incident and to no other tasks. More general correspondence with k > 2 can now be identified as hypergraphs, a generalization of linear graphs studied by C. Berge and others (as discussed e.g. in C. Berge Graphs and Hypergraphs, North-Holland 1976). In a hypergraph an edge becomes an unordered subset of vertices rather than a pair of vertices as in a linear graph. Thus the hypergraph is exactly the proper mathematical object for representing a multiple correspondence.

Present research in assigned availability is directed toward the study of full service probability in the restricted k = 2 case. One approach is to investigate the use of cycle structure in the double correspondence graph and use that structure to compute the desired measure. This direction of research is motivated by the fact that service is lost only when there is no cycle in a connected component. The cycle structure in a graph is complex and it is felt that some use must be made of cycle bases. A cycle basis is generated by a spanning subtree. Thus the research activity has been concerned with the study of spanning subtrees, cycle bases, and related concepts. In particular the generation and enumeration of these quantities is of interest. There is a rich literature on this subject as discussed e.g. in Combinational Algorithms by Reingold, Nievergalt, and Deo. However, some quantities of interest have not been found in the literature and have required the development of appropriate algorithms as illustrated below.

Work on the methodology discussed above has suggested that not all cycle bases act in the same way in regard to the assigned availability model. It may be that some basic cycles are more important than others

proven elusive. Present research has made some studies of this question with complicated and inconclusive results. One thing that seems to emerge is that the connection may be closer to dam theory than to queuing because the non-cascade property seems to be difficult to implement within a queue model in a way that leads to the deterministic model. Further study of this theoretical issue will improve understanding of the model and its application to problem areas. It may indicate useful alternative model developments as well.

2.2 Assigned Availability

Let U denote a set of n similar operational units (e.g. components) and T a set of m similar tasks or requirements (e.g. operations within a hardware system configuration). A correspondence between the elements of U and T that associates each element of U with k specific elements of T is called an arrangement. If $k \neq 1$ the arrangement is a multiple correspondence.

The concept of multiple correspondence was first introduced by B. O. Koopman at the First International Conference on Operational Research in 1957. He suggested them as a basis for mathematical models of a variety of military, industrial, and scientific applications. Early work on the general nature of multiple correspondences was carried out by Koopman, in his Proceedings paper, by P. D. Finch in a 1958 paper, and by C. Marshall in a series of three papers in the early 1960's.

Present research by C. Marshall in this area consists of employing the multiple correspondence model in the study of assigned availability. After the associations between units and tasks has been made some units are removed at random (fail). In the current model this removal (failure) has a constant probability p for each unit, independent of all other units. The remaining units are available to be assigned to any task with which they are associated by the multiple correspondence. A unit can be assigned to one and only one of its associated tasks. The major question is: how well the given correspondence will be in supplying units to tasks? This gives a model of system availability. Two measures of performance were used in the early studies of multiple correspondences and both continue to provide measures of assigned system availability. One measure

3.3 Research Reports (continued)

Antoine Gerard Cormier, "A Quantitative Analysis of the Effect of Organizational Structure on Software Engineering Management, 1983.

R. W. Schmidt, "Fitting an Exponential Software Reliability Model to Field Failure Data", 1983.

George Estes, "Algorithmic Complexity Estimation for Computer Software", M.S. Thesis in Computer Science, June 1982.

Howard Hwang, P.C. Whang, and Martin L. Shooman, "Static and Dynamic Loads on Nuclear Power", Brookhaven National Labs Research Report, 1983.

Martin L. Shooman, "Independence Bounds and Approximations for System Availability Computations", Polytechnic Research Memo No. 1, April 6, 1984.

Paul E. Janusz, William R. Turoczy (with contributions by Martin L. Shooman), "Application of Software Test Tools to Battlefield Automated Systems, Picatinny Arsenal, April 1984.

4.0 Professional Activity

- Clifford Marshall continues as an Associate Editor of Naval Research Logistics Quarterly.
- Leonard Shaw

  Appointed Permanent Head of Department of Electrical Engineering and Computer Science - January 1983.

  Elected Fellow of IEEE for contributions to modeling, estimation and control of stochastic systems - January 1984.

  Program Chairman of June 1984 American Control Conference.

  Administrative Committee of IEEE Education Society 1983,4.

  Fellows Review Committee of IEEE Education Society 1984.

  Member of IEEE Publications Board and Chairman of Finance Subcommittee.

  Member IEEE Budget Development Committee.

  IEEE Control Systems Society Publications Chairman - promote and review proposals for IEEE Press books.

- Martin Shooman

  Awards

  P. K. McElroy Award for Best Technical Paper at 1983 Annual Reliability and Maintainability Symposium. (Three previous such awards in 1968, 1971, 1977)

  Consulting

  Grumman, Calverton, "Hardware Fault Tolerance", March 1982.

  Electric Power Research Institute, "Reliability Review of GE Turbine Redundant Microprocessor Controls, Aug. 1982.

  Burrows, "Software Reliability", Aug. 1982.

4.0 Professional Activity (continued)

o Martin Shooman (continued)

Consulting (continued)

Brookhaven National Labs, Reliability of Nuclear Containment Structures, 1978 - 1983.

Control Data, Expert Witness on Software Errors, Fall 1982.

Gould, Melville, N.Y., Availability of Air Traffic Control Systems, Summer 1983.

Norden, Norwalk, Conn., Corporate Review of Software Engineering and Design Procedures, Summer and Fall 1983.

U.S. Army, Picatinny Arsenal, Dover, N.J., Automated Software Testing, Fall 1983, Spring 1984.

Administrative Positions

Director, Division of Computer Science, Jan. 1981 - Jan. 1984.

Associate Director, Office Automation and Software Engineering, Center for Advanced Technology in Telecommunications, 1983 - Present.

Professional Societies and Activities

Member of IEEE Long Island Section Fellows and Awards Committee, 1979 - Present.

Member of IEEE Edison Medal Award Committee, 1980 - 1983.

Member of IEEE Awards Board, 1984.

Chairman, Brookhaven National Labs Study Committee on Nuclear Power Plant Structural Loads, 1981 - 1983.

Member of Executive Committee of the IEEE Computer Society Technical Committee on Software Engineering.

University member of the National Security Industries Association (NSIA) Committee on Software Engineering.

Advisory Board Member, Annual Reliability and Maintainability Symposium, 1984.

Chairman, LIFT Symposium, "Software Update", Melville, N. Y. May 24, 1984.

Chairman, Planning Committee, IEEE/ACM/POLY "Conference on Software Tools", Scheduled for NYC, April 1985.

END

FILMED

9-85

DTIC